The Metafora Design Principles for a Collaborative, Interoperable Learning Framework

Andreas Harrer, Thomas Irgang, Andreas Lingnau, Norbert Sattes, and Kerstin Pfahler

Catholic University Eichstätt-Ingolstadt, 85072 Eichstätt, Germany
{andreas.harrer,thomas.irgang,andreas.lingnau, norbert.sattes,kerstin.pfahler}@ku-eichstaett.de

Abstract. In this paper we present the Metafora project for the support of collaborative learning activities in larger problems of science and math topics. We will present the design principles that guided our technical development of an architecture supporting collaboration across different learning tools. Interoperability between the tools mediated by so called referable objects is described, as well as the design issues of awareness and visualisation for the learning groups. We demonstrate the flexibility of our designs and framework in giving example cases for the usage of the Metafora framework with different tools and educational scenarios.

Keywords: Web-based collaborative applications, collaborative workspaces, computer-supported collaborative learning (CSCL), design principles.

1 Introduction - Web-Based Collaboration in Metafora

Over the last decades several collaborative learning environments have been developed for the classroom. The technology used to realise those systems slightly changed but there are still many pros and cons for either web-based or non web-based applications. As web-based systems usually imply restrictions compared to client based applications when it comes to flexibility and adaptivity of user-interfaces, web-based applications are more or less ready to use when opened in a browser.

In the EU funded project Metafora[1] we are bringing together a set of different learning tools for science and math that have been developed in other context and projects. Metafora combines different pedagogical strands, namely constructionism and collaboration, resulting in an approach called *learning to learn together* (L2L2) [1]. Constructionism [2] stresses an active role of the learner who is (re-)constructing knowledge by herself instead of knowledge being delivered by the teacher. Usually this is achieved by direct construction of artefacts,

[1] The Metafora project is co-funded by the European Union under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme for R&D (FP7), Contract No. 257872, http://www.metafora-project.org/

P. Antunes et al. (Eds.): CRIWG 2013, LNCS 8224, pp. 192–207, 2013.

models, programs etc. Collaboration is another facet to engage students to a more active attitude during learning, stimulating argumentation, negotiation, planning and different kinds of strategic skills referring to management and task solution. This approach is applied to science & math classroom activities of approx. 20 school hours for project groups of 3-6 pupils of ages 12-16. The learning tools are integrated within the browser-based Metafora framework for collaborative and self-regulated learning and organization of the learning process to allow students to make use of the full potential of these tools while working within one integrated web-based environment, using a single login and having central access to all the available tools. Among these tools are constructionist Microworlds for Math and Physics, game-like environments for sustainability (Sus-X, cf. section 5.1) and ballistics (PIKI, cf. section 5.2), and the eXpresser editor for constructing mathematical patterns and algebraic equations [3]. As an umbrella Metafora provides a rich collaborative environment with a shared planning space [4], and – shown in Fig.1 – a group chat and the LASAD argumentation environment[2] as the selected tool tab in the main part on the right side.

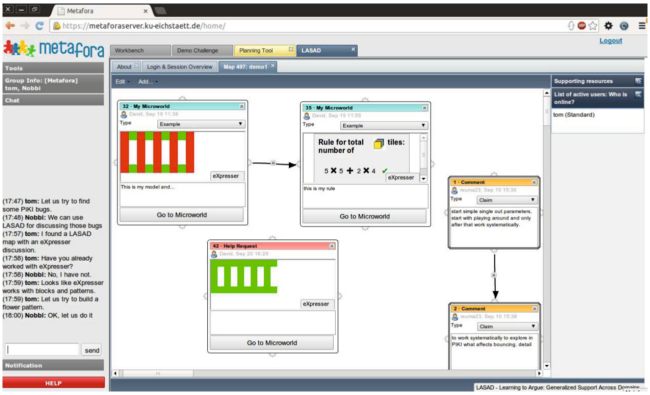

Fig. 1. Metafora system with the LASAD tool visible in the main part

One challenge of web-based collaboration is to overcome a major communication deficit between client and server. While standard web-based interaction is based on the request–response schema, i.e. a web client sends a request to a web server and the server sends a response back, this can only be used to notify a server of a user's action. Other users working on different web clients need to be notified of the same action by the server. One way to realise this update is to change the request–response mechanism and use the so-called *server push* technology which enables the server to actively send messages to web clients. Metafora uses this technology for the distribution of information between the web-based clients: the built-in chat, the graphical *Planning Tool*, and the propagation of updates in the awareness and sharing tool, called *Workbench*.

[2] http://cscwlab.in.tu-clausthal.de/lasad

2 Distributed Architecture and Collaboration Framework

Since the Metafora system is intended as a framework for various learning tools, a flexible approach for integration of learning tools is required, as well as a concept how to support learners appropriately in their process by means of intelligent analysis components.

Our resulting proposal for Metafora's architecture can be seen as a modified *blackboard architecture* [5]: several learning tools and analytic components can be used at the same time and do not interact directly with each other (loose coupling). All analytic components will be able to read and take up results from other analytic components and all learning tools will be able to send commands for interoperation between the tools, because a unified language is used between all these components. This allows the required flexible combination of learning tools and associated domain-specific and tool-specific analytic components with cross-tool or domain-independent components and can also be used with already existing indicator-based analyses, as e.g. from the math microworld eXpresser [3].

Our technical implementation to achieve this is to use a well-known and stable communication infrastructure, the *eXtensible Messaging and Presence Protocol* (XMPP), which for example is also used for Google Talk, the Facebook Chat and Whatsapp. Because of the wide spread use of the XMPP technology there are libraries for all common programming languages and frameworks. This is important for Metafora because it allows to easily integrate existing learning tools and microworlds into the framework.

For Metafora we need an open and flexible inter-tool communication between the various tools that do not require knowledge of other existing tools according to the principles of blackboard architectures. For this end we use the extension XEP-0045: Multi-User Chat [3] (MUC) to support multiple clients. We also want semantically interoperating communication between the various tools to generate a better learning experience. Even if XMPP is already an XML protocol we use it only as a (replaceable) communication channel. The messages sent via XMPP are represented in a unified format, called CoLoForm [6], to allow a uniform interpretation across the different analytic components, regardless of where the feedback / analysis was produced. The advantage of using CoLoForm as communication XML format and XMPP only as communication channel is the perfect fit of CoLoForm for our needs and the independence of the XMPP protocol and its implementations. To reduce the message overhead we use three MUCs for Metafora. The analysis MUC contains messages from analysis components and landmarks from microworlds. The command MUC contains the inter-tool communication, which means for example feedback commands which the framework should display for a set of users. The logger channel contains actions for most of the user interaction with the system or components of the system and allows creation of a joint log database. Cross-tool analysis

[3] http://xmpp.org/extensions/xep-0045.html

components can subscribe to both log- and analysis-channel and will post their results to the analysis-channel. The biggest advantage of this architecture is that any tool, which can be extended, can interact with the Metafora framework. Figure 2 shows the overview of the Metafora architecture including the blackboard-like approach for the analytic subsystem.

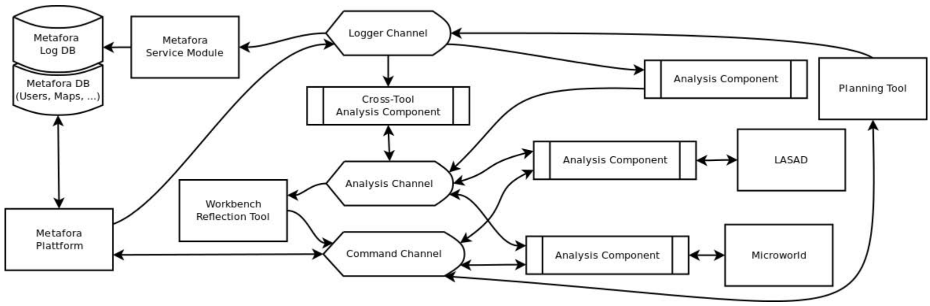

Fig. 2. Architecture overview of the Metafora System

Microworlds and learning tools with a web-based graphical interface can be integrated into the Metafora GUI with the help of inline frames. The technological basis of the tool doesn't matter. The Metafora framework also allows opening of any URL as inline frame with a command message. Most of the currently integrated microworlds use the Unity framework[4], which is a client-only framework like Adobe Flash, and these microworlds use a direct XMPP connection to our server for logging and inter-tool communication. For these learning tools, that do not have their own collaboration server, the file storage feature of the Metafora platform is especially interesting: allowing the client to upload states of the learning workspaces and retrieving them afterwards allows the usage of the tools inside the context of a larger learning activity without the need to create a collaboration server for each tool.

The learning tools for collaborative usage like LASAD[5] and PlanningTool are client-server based and use the *Google Web Toolkit*[6] as technological basis. Both LASAD and the planning tool use only a XMPP connection from the server side component and push the collaboration events to the clients.

The architectural backbone of the Metafora framework with its three channels of communication is fully implemented, while analysis components are currently under development. In this paper we will focus on the cross-tool interactions mediated through the command channel.

---

[4] `http://unity3d.com/`

[5] `http://cscwlab.in.tu-clausthal.de/lasad/`

[6] `https://developers.google.com/web-toolkit/`

3 Design Principles for Multi-user Interfaces

Group awareness [7] is a major issue for designing collaborative user interfaces. The goal is to raise a user's awareness of the activities conducted by the other users of the group and to provide an orientation of the social context in which the system is used by the group. Depending on the degree of joint environment and the nature of the collaborative / cooperative task, awareness can manifest itself differently. [8] gives an overview about mechanisms how to support awareness in synchronous vs. asynchronous scenarios and closely vs. loosely coupled task contexts. In our Metafora case of supporting small groups in synchronous collaboration on learning to learn together, we can focus mainly on the synchronous and closely coupled case. We use a relaxed WYSIWIS ('what you see is what I see') approach in a way that the users can have their own view on a joint planning map and also different tools in focus currently. This means that sharing a focus of attention has to be supported by the system design. Among the support functionalities to provide awareness of peer learners' actions we provide:

- awareness of incoming chat messages by giving visual cues, when the user's chat is folded away in favour of other tools
- awareness of incoming notifications sent by intelligent system components or learning facilitators in a similar way to the chat awareness
- display of notifications of different severeness using three levels of intrusiveness (non intrusive, transient messages, model pop-ups)
- overview of group composition, online status and shared resources
- process awareness of the groups activities and achievements over time. The representation designed for this is currently called 'reflection tool' and will be described in detail in the following paragraphs

One other principle for the design of user interfaces for collaborative learning activities that is particularly relevant for long-term and complex activities involving multiple and intertwined tools, such as in Metafora, is a design for visualisations that do not put too much load on the user, yet still keeping the potential to retrieve rich information, if requested by the user. Our approach will be oriented towards the principles known as the 'visual information-seeking mantra' [9] that is cited as 'Overview first, zoom and filter, then details-on-demand'. The designs and proposals we will present for reflection on learning activities in subsection 3.2 will follow this mantra, in giving first an overview that provides visual cues of potentially relevant areas, then interactive zooming, re-focusing and filtering techniques to explore parts of the visualisation in detail.

Finally, in a highly heterogeneous environment as the Metafora system with several distinct learning tools, support tools, and analysis components, a real danger of media breaks when changing between different tools exists. Thus, there is a need to carefully consider aspects of interoperability, seamless transition between tools, and coherent interaction flow between the tools. In the next subsection 3.1 we will introduce the concept of *referable objects* that are one proposal in Metafora to allow the seamless navigation of users between different tools with a uniform mechanism inspired by theory, namely the concept of *boundary objects*, and the technical concepts of hyperlinks in the web and references to objects in programming languages.

3.1 Referable Objects as Boundary Objects for L2L2

Referable objects are not a new concept, but rather were inspired by the concept of boundary objects. Boundary objects are coined by Star and Griesener, who originally used boundary objects to maximize both the autonomy and communication between worlds in their study from 1989. They defined them as objects to depict the divergent use of information in divergent groups. Although they are plastic, a defining characteristic of boundary objects is that they contain enough changeless content to preserve a global identity [10]. In 2004 Hoyles used boundary objects in a classroom context. She wrote "a boundary object provides a generalized mechanism for meanings to be shared and constructed between communities" [11]. We will take up the concept of boundary objects as so called referable objects serving as mediating artefacts between different workspaces explained in the next paragraph.

We found a need for boundary objects in Metafora because students need to share individual elements, or objects, from one tool in another tool to allow discussion about the workings of one tool within another. When a group can share these objects (also with other groups), it is then possible to use the most appropriate tool to discuss and evaluate ideas focused on these objects, rather than being tied to the tool from which the object originates. This ability to visualize and discuss an individual object from one tool within another is powerful and helpful to students, but there is an obvious weakness: viewing an object isolated can cause the loss of context and meaning compared to viewing the object within its original environment. To alleviate this issue, and allow any user to see the full context of the given object, a **reference** in form of an address to the source of a tool where the object was created, is also recorded with the referable object. Through this reference, that has also a visual preview attached to it (i.e. a thumbnail picture of the object in context to allow identification of the object across different tools) learners are able to navigate directly to the place of the element and inspect it. So we can see that our concept of referable objects expands upon the concept of a boundary object, and as such we offer the following definition of a **referable object**: "an individual element or product of any tool that is recorded by the system in such a way as to allow students, teachers and researchers to reference this object for discussion or evaluation at a later point in time."

Referrable objects can be shared as objects to discuss and review in various places of the Metafora system. The students' ability to "discuss in the appropriate place" is one of the subconcepts of L2L2 that shall be supported by the system: thus, both formal argumentation in the LASAD system and more informal or coordinative talk in the group chat can be a place to share referable objects in. The screenshot in Fig.3 shows a referable object shared with the group in both the chat and the LASAD system. In both spaces multiple stakeholders, such as group members, teachers, or peer groups, can follow the reference and

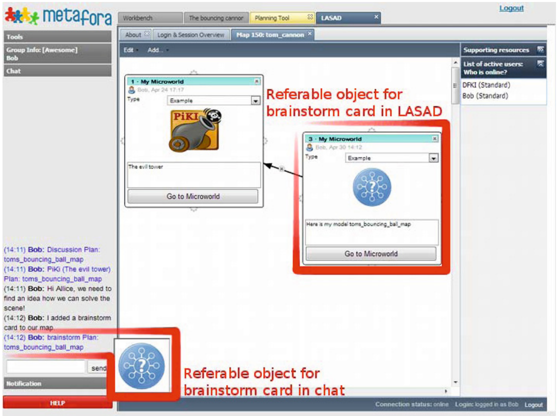

Fig. 3. A brainstorming activity shared from a planning map as a referable object to group chat and the LASAD argumentation system

directly inspect the shared object in its direct usage context, here the specific brainstroming activity in the planning map.

3.2 The Reflection Tool – Combining Principles of Awareness and Visualisation

One of the key factors for learning to learn together in Metafora is the ability to self-regulate, co-regulate and to reflect on others' and own actions during the learning process. In early experimentation we found out that the planning tool is frequently used not only for planning, but also for documenting and reflecting on the current progress and state. To stimulate these very much desirable meta-cognitive processes, we co-designed with some input from our pedagogical partners a visualisation for the support of reflection processes, shortly 'the Reflection Tool'.

Among the information the representation should provide, we identified the following situations and indicators:

- Help requests produced by the members of the group
- The extent to which these requests were responded and by whom, inside the group as well as from the outer community
- How close the group is to reach its goals when comparing plans to enactment
- To what extent the group carried out reflection over their activities
- If the group was confronted with a lot of disputes
- If the group reached consensus at any point

This information could be presented in isolation in table form or some time series diagram as prevelant in current *learning analytics* representations, such as for example the Learning Dashboard [7]. Yet, we believe that on the one hand the representation should be cognitively linkable to the tools, the information originated from, and on the other hand that the concrete visualisation should be designed according to sound information visualisation principles that don't require expertise in 'reading diagrams' as expected from business analysts and statisticians.

On the one hand we follow the argumentation in [12] to embed analytic features into the learning tools to contextualise it with the learning activity, on the other hand we sought inspiration in real-life and everday representations for the concrete visualisations: Based on an analogy of sports tickers (e.g. football, hockey, basketball, rugby) we connect a temporal dimension with the relevant events (goals, ejections, substitutions) happening in relation to timepoints and activities. Our events are meaningful L2L2 situations as listed above, while the time dimension is additionally enriched with information about planning activities happening over time. Each card in the planning tool is represented as a bar scaled according to the duration of the activity associated with the card, allowing an inter-linked interpretation of the reflection tool and the planning tool by the user. The visual representation chosen here shares some features with both Gantt diagrams from project management and also from the representation of a 'down' in American Football tickers. An example presentation of the reflection tool in conjunction with the relevant plan is shown in Fig.4.

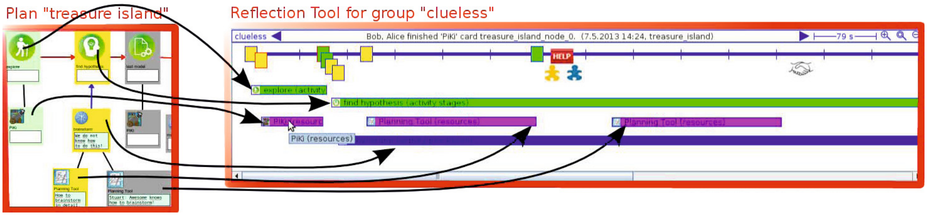

Fig. 4. Reflection tool with iconic representations of important learning events

The picture sketches out how reflection can be supported by means of an interactive visual representation that shows how the students acted and interacted across a temporal dimension shown from left to right in a time axis. Along this line the students can perceive their main activities represented as bars stretching out for their duration, here a green activity stage card started inside of which a resource card (i.e. a tool workspace) has been created and used. Relevant learning events, such as reaching intermediary milestones (shown as yellow and green cards), help seeking (the yellow user asked for help and the blue responded to it) or reaching a consensus (handshake symbol), are represented in relation to their temporal occurrence. The representation provides many interactive features to focus on specific elements, using filters and zooming features, to get overview and

[7] http://learningdashboard.org/

focusing capabilities in tune with sound visualisation principles from the HCI field: similar to the methods in image processing tools or video editing a user can zoom in by marking a rectangle defining the interval to zoom in to. Hovering on the iconic representation of a learning event gives more details, while clicking on it can be combined with an auto-focusing and centering of this event in the temporal dimension.

Since the reflection tool has been designed very recently in a participatory process based on users' needs and preliminary results of our evaluation, the tool has not yet been formally evaluated. We are currently in the process of designing controlled studies that specifically answer research questions about the support of the self- and peer-reflection processes and usability issues with the reflection tool. This is a first step, because the specific evaluation in-vivo will be difficult because of the complexity of the pedagogical scenarios of Metafora and also the system complexity involving multi-tool actions. In a second step we want to explore the role, usage, and aptitude of the reflection tool inside the whole Metafora approach, by adding specific questions on that to our final main experiments in class.

4 Example Case of Collaboration Principles

In this section we give an example how the Metafora framework can be used. One of the central components of Metafora is the Planning Tool. With the help of this component the students collaboratively plan their approach to the learning scenario and get a quick overview of open and running tasks.

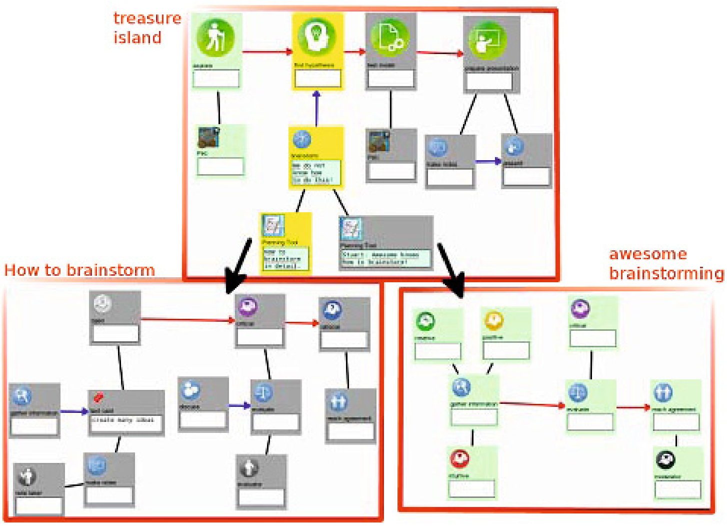

Fig. 5. Alice's and Bob's plan treasure island and both referred brainstorming plans

In our example the two students Bob and Alice get the task to solve a ballistic game scene in PIKI with the minimum number of shots using different bouncing surfaces. Therefore they join a group and create a new plan *treasure island* to approach the task. A screenshot of this plan is shown in the top of Fig.5. First they want to explore the plan to get an idea how the scene can be solved. Afterwards they want to phrase their ideas using brainstorming, test there conclusion and present the ideas to the other students. To explore the scene, they added a PiKI resource card which represents a link to a PiKI workspace. If they consider their playing with PIKI as a success they can checkmark the activity, thus changing the background colour of the card to green. Now they want to find an idea how to solve the scene with the help of brainstorming, but Bob and Alice are clueless how brainstorming works. To overcome this lack of knowledge they decide to create a new plan *how to brainstorm*. They could just create a new plan but they use the Wiki-like feature to add a new Planning Tool resource card. Now they can start using the card to create and link a new Planning Tool plan to this card. Everyone can access the new plan *how to brainstorm* through this Planning Tool resource card from *treasure island*. Bob and Alice can now build a visual model based on their ideas. This plan is also contained in Fig.5 lower left.

When they finish their plan, they are not sure if their brainstorming model is valid and decide to ask for help. The Metafora framework supports asking the own group or others for help. They decide to do a general help request and Stuart, who is member of the group *awesome*, reads their request and decides to help. So he uses the group info panel to switch to group *clueless* and chat with Bob and Alice. Stuart knows of an old brainstorming plan formerly created in the group *awesome* and decides to share this plan with the group *clueless*. The Planning Tool features sharing of plans with other groups. This means other groups get direct access to these plans through the plan selection options. Stuart uses another way to share the plan. He adds another Planning Tool resource card to the *treasure island* plan and links it to the plan *awesome brainstorming*. The Fig.5 contains this plan on the lower right.

Now Stuart can switch back to the group *awesome* and Bob and Alice now have two models for brainstorming. They can revise their model with the help of Stuarts plan and continue with the learning scenario.

5 Practical Usage of the Framework and Integration of External Learning Tools

When developing a technical framework, one of the questions to be answered is if the framework facilitates the re-use and integration of new components. Thus, one hypothesis we wanted to put to the test is if our framework and architecture allows the integration of different types of learning tools and saves time / effort compared to integrating tools directly without a framework.

To test the practical applicability of our Metafora framework, in this section we will present the modifications that are necessary to integrate existing learning tools and some design issues for new tools. First, we will introduce the functionalities that can be used to integrate learning tools semantically and then we present two showcases of successful integration: the first is a pre-existing, formerly not web-based, microworld in the domain of sustainability, the second is a learning game that has been developed from scratch at a time when our technical interfaces had been thoroughly specified already.

Among the important *scenarios* of cross-tool interaction mediated by the platform we consider the following situations / scenarios:

1. creation of a new workspace in a tool by means of the creation of a resource card in a planning map
2. creation of a referable object in a learning tool and making it referable in a space of discussion
3. inspecting a referable object by following the respective reference; this is technically realized by opening a specifically parameterized URL that contains the information to re-create or directly inspect the referable object
4. storing a learning tool model or state into the platform, a feature mainly used by serverless, client-only tools
5. retrieving a learning tool model or state from the platform and using this in the tool where it originated from
6. sending notifications, feedback messages, and analysis results

All these scenarios are triggered by sending XML messages of a well-defined format via the XMPP communication infrastructure. Tools that are able to support these functionalities subcribe to the XMPP command channel and take up the messages, that are relevant for them to be executed. The approach of defining an API for learning tools to react to specific external commands is similar to the 'remote moderation API' [13], yet, our approach is completely transparent to different programming languages and approaches, as long as XMPP is supported, while the earlier approach was using java interfaces, thus being confined to one programming language.

We will demonstrate both the interactions between tools and platform, messages exchanged during this interaction, and the necessary implementations for the two learning tools presented in the next subsections.

5.1 SuS-X, Integrating a Non Web-Based Legacy System

SuS-X is a game template that supports learning in the domain of sustainability. It is a microworld for non technical users, to play and design their own game. The game can be designed by adding content, sites and site properties to the game template. The rules of the game and the end of the game can be defined by setting appropriate values. Thus users can explore, change and reconstruct a model which will then be carried out by players. Figure 6 shows *SusCity*, which is an example microworld template of *SuS-X*, embedded in the Metafora platform, in which users can simulate and experience a sustainable way of living by making a trip in a designed city.

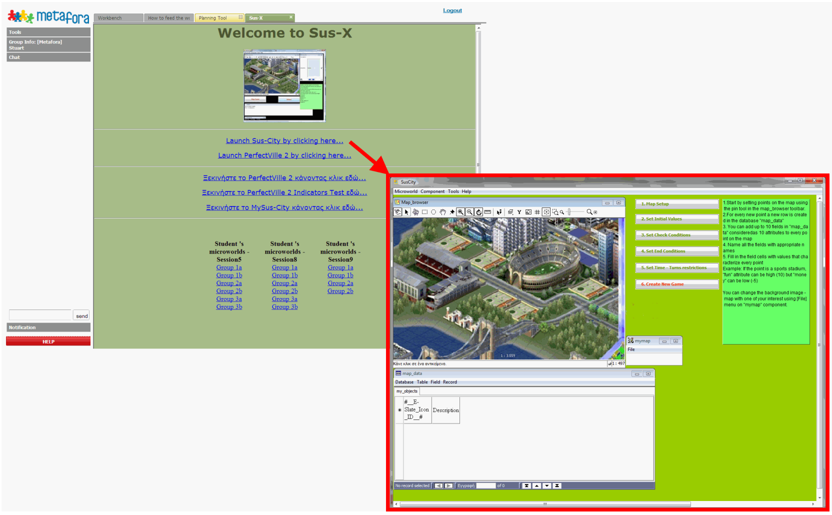

Fig. 6. Intermediate web-page launching Sus-X, here an instance of SusCity

As a pre-existing, client-side and non web-based tool of Metafora developed by our project partners from the Educational Technology Lab (ETL) in Athens, *SuS-X* had to be adapted to the framework. To support the heterogeneity caused by its initial implementation, it was integrated as a standalone application, launchable via an intermediate web-page which is hosted on a server. If *SuS-X* is started through the Metafora framework a website is opened which offers URLs to launch different *SuS-X* learning environments on the client PC. This is realised through an applet which installs the *E-Slate* platform, the technological basis of *SuS-X*. After the installation *SuS-X* is launched, as shown in Figure 6. E-Slate is a learning environment, enabling the visual manipulation of pre-fabricated software elements.

Technically, *SuS-X* is implemented as a Java application, which uses the XMPPBridge with the Smack library offered by the author team. Through this XMPPBridge a *SuS-X* instance launched through a resource card connects to the XMPP channels of Metafora, creates a reference to the workspace (e.g. scenario 1 from the list above) and writes logging information for the analysis components of Metafora (scenario 6). *SuS-X* will also offer a mechanism to store and retrieve saved states, which is currently under implementation (scenarios 4 and 5). If a user decides to save his current work, the microworld will create a file with the modifications and upload it into a database via a standard HTTP fileupload. The fileupload returns a unique ID to this file, through which the stored file can be retrieved at any later point in time. This mechanism will be explained in more detail in the next section.

Although *SuS-X* is a non web-based client application, for the user it appears seamlessly integrated into the Metafora platform and allows interoperable communication with the other tools as well as transitions from and to them.

5.2 PiKI, Integrating a New Unity-Based 3D Microworld

The microworld *Pirates of the Kinematic Island* (PiKI)[8] is an educational game in a setting with pirate flavour which allows students to experiment playfully with parabolic trajectories. PiKI supports two game modes: Students can build scenes where treasures are protected with obstacles and publish this scene as a challenge for other students. Other students can play a published scene and try to collect the treasures with the cannon ball by setting velocity and launch angle of the cannon ball and add surfaces with different bounce characteristics.

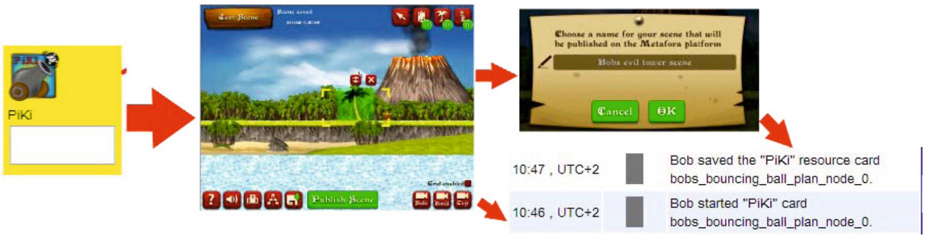

Fig. 7. Creating a new PiKI scene - users' view

PiKI[9] is one of the integrated Unity based microworlds and can be used as a none-collaborative stand-alone game. Through the integration into the Metafora framework most of the none-collaborative disadvantages can be overcome. When, for example, Bob and Alice want to learn something about parabolic trajectories with the help of PiKI they can log into Metafora and create a new plan. Then Bob can add a PiKI resource card and starts using the card (Fig.7 on the left). The Planning Tool sends a command XML message to the framework with the PiKI URL and plan informations (scenario 1 in the list of tool interactions). The framework adds the user informations to the URL and opens it as inline frame on Bobs client (scenario 3). Now Bob and Alice can build a new PiKI scene together if they work on the same client. The scene is stored and refreshed after every change to the Metafora version storage (scenario 4). While creating the scene PiKI logs the users' actions to the XMPP logger channel and generates analysis messages (scenario 6, and Fig.7 lower right) if the scene was changed. After publishing the scene Bob and Alice can split and try to solve instances of this scene, which is now available to all Metafora users.

Fig.8 shows an overview of the flow when Alice and Bob play the scene in parallel: Alice can add another PiKI resource card to their joint plan, start PiKi and select the created scene for playing while Bob can use the old PiKI card. Technically the stored scene is requested from the Metafora storage in its

[8] PiKI has been developed by our project partner Testaluna in co-design with pedagogical input from other project partners; in this paper we focus on the integration aspects that have been coordinated by our author team.

[9] `http://test.silentbaystudios.com/metafora_piki/test/piki.php`

current state and retrieved. The card is now connected with an instance of the scene (which means that – according to scenario 2 – the scene is now a referable object and the resource card the reference to it) and PiKI sends analysis messages for the new instances. These messages are visible for all students in the same group (again scenario 6 - notifications). Actions during the game are logged to the logger channel and can be evaluated by every attached analysis component. Analysis components can react with feedback messages for Bob or Alice (scenario 6 - feedback). These messages are displayed to the student through Metafora and PiKI. If one of the players hits a treasure PiKI sends a landmark message (scenario 6 - analysis) to the analysis channel, which is shown in Fig.8 in the lower middle. If Bob or Alice want to resume later they can log out from Metafora and resume with their current game state through their resource card (scenario 5). If one of them has problems with the solution or they want to discuss their solutions they can add a LASAD resource card to their plan and start using the LASAD discussion tool (scenario 1, but now with LASAD). Now they can share their PiKI instance as a referable object to LASAD (Fig.8 lower right). To do this, PiKI sends a command message which is picked up by LASAD and triggers the creation of a new discussion node with a reference to the PiKI instance (scenario 2). With this discussion node the saved state of the PiKI instance can be opened using scenario 3. This is also realised with an *open URL* command message.

Fig. 8. Playing a PiKI scene - overview

This example shows how easily even none-collaborative, client-only microworlds like PiKI can be integrated into the Metafora framework and the benefits of this integration. For the integration of PiKI into Metafora a Unity native XMPP library, agsXMPP[10], was used. Since PiKI has been developed from scratch during the Metafora project, all the technical protocols for scenarios 1–6 had already been specified. Thus, the purely technical integegration was achieved with relatively little effort of – to our estimation – a few days of programming. The conceptual design of the granularity of referable objects, appropriate storage formats etc. is considerably higher, but independent from the technical framework and necessary anyway for each learning tool to fulfill its educational purpose.

[10] `http://www.ag-software.net/agsxmpp-sdk/`

As the above scenario showed, PiKI can interact with other integrated tools like LASAD and benefit from more general analysis components, such as cross-tool analysis components that combine information from planning and problem-solving behaviour. With the help of the Metafora version storage it was even possible to transfer the current state of this client-only microworld to other users and clients, using the Metafora storage as a collaboration server.

6 Conclusion and Further Work

In this paper we presented the web-based collaboration framework Metafora and discussed design principles that guided our development of the software framework and the learning tools to be integrated. On the conceptual level we introduced *referable objects* as specific boundary objects that allow communication between peer learners and seamless transition between different learning tools. On the technical level we presented a collaboration architecture with technical interfaces that supports integration of external learning tools also on a semantically interoperable level. To support learners in their reflection about self-regulated learning processes, we designed a tool that integrates awareness and visualisation principles in connection to the collaborative planning tool. This is a specific case of embedded Learning Analytics features [12]. We showed the usage of these design principles with an example from practical usage of referable objects and resource cards in a Metafora learning activity. Additionally, we presented two cases of technical integration of learning tools using our framework's interfaces. These cases supported our hypothesis that a broad variety of learning tools, even non web-based or plugin-based, can be integrated successfully into the Metafora system. The flexibility of our framework has been proven by the cross-tool interoperability achieved when following our technical interfaces and by the possibility to allow client-only learning tools to store and retrieve states in combination with referable objects.

We are currently evaluating the educational effectiveness of our design principles by means of qualitative studies both in-vitro and in-vivo, i.e. in controlled lab settings and in school classrooms. Eye-tracking technology has been used for the lab studies to explore the usefulness of our awareness features, while school classes have been using the system successfully now for 18 months in various educational setups. To create sustainability of the system and invite the integration of additional learning tools into our collaboration framework the platform of the Metafora project is published as Open Source at `https://github.com/metafora-project`.

References

1. Dragon, T., Mavrikis, M., McLaren, B.M., Harrer, A., Kynigos, C., Wegerif, R., Yang, Y.: Metafora: A web-based platform for learning to learn together in science and mathematics. IEEE Transactions on Learning Technologies (preprint, 2013)
2. Harel, I., Papert, S. (eds.): Constructionism. Ablex Publishing, Norwood (1991)

3. Gutierrez-Santos, S., Mavrikis, M., Magoulas, G.D.: A Separation of Concerns for Engineering Intelligent Support for Exploratory Learning Environments. Journal of Research and Practice in Information Technology 44(3), 347–360 (2012)
4. Harrer, A., Irgang, T., Sattes, N., Pfahler, K.: SoCCR – optimistic concurrency control for the web-based collaborative framework metafora. In: Herskovic, V., Hoppe, H.U., Jansen, M., Ziegler, J. (eds.) CRIWG 2012. LNCS, vol. 7493, pp. 153–160. Springer, Heidelberg (2012)
5. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: A System of Patterns. John Wiley & Sons, Chichester (1996)
6. Harrer, A., Martinez-Mones, A., Dimitracopoulou, A.: Users' data: Collaborative and social analysis. In: Balacheff, N., Ludvigsen, S., de Jong, T., Lazonder, A., Barnes, S. (eds.) Technology-Enhanced Learning – Principles and Products, ch. 11, pp. 175–193. Springer, Berlin (2009)
7. Borghoff, U.M., Schlichter, J.H.: Computer-Supported Cooperative Work: Introduction to Distributed Applications. Springer, Berlin (2000)
8. Fuchs, L., Pankoke-Babatz, U., Prinz, W.: Supporting cooperative awareness with local event mechanisms: The groupdesk system. In: Marmolin, H., Sundblad, Y., Schmidt, K. (eds.) Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work, ECSCW 1995 (1995)
9. Shneiderman, B.: The eyes have it: A task by data-type taxonomy for information visualizations. In: Proc. Visual Languages, Boulder, CO, September 3-6, pp. 336–343. IEEE Computer Science Press, Los Alamitos (1996)
10. Star, S.L., Griesemer, J.R.: Institutional ecology, 'translations' and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907-1939. Social Studies of Science 19(3), 387–420 (1989)
11. Hoyles, C., Noss, R., Kent, P.: On the integration of digital technologies into mathematics classrooms. International Journal of Computers for Mathematical Learning 9(3), 309–326 (2004)
12. Wise, A.F., Zhao, Y., Hausknecht, S.N.: Learning analytics for online discussions: a pedagogical model for intervention with embedded and extracted analytics. In: LAK 2013 – Learning Analytics and Knowledge, pp. 48–56 (2013)
13. Harrer, A., Ziebarth, S., Giemza, A., Hoppe, U.: A framework to support monitoring and moderation of e-discussions with heterogeneous discussion tools. In: Diaz, P., Kinshuk Aedo, I., Mora, E. (eds.) The 8th IEEE International Conference on Advanced Learning Technologies, ICALT 2008, pp. 41–45. IEEE Computer Society, Los Alamitos (2008)